

In the Claims

Claims pending:

- At time of the Action: 1-26.
- After this Response: 1-26.

Currently Amended claims: 1, 7, 13, 18, and 21.

Canceled or withdrawn claims: None.

This listing of claims replaces all prior versions and listings:

1. (currently amended): A method comprising:

receiving a request for a Web page;

identifying an Active Server Page associated with the requested Web page, wherein the Active Server Page includes a compiled user interface template created using an Active Server Page Language, which when compiled is executed through an application programming interface developed using a system language to generate the requested Web Page in the system language from the user interface template created using the Active Server Page Language such that the user interface template uses a separate application programming interface from the application programming interface that is configured to use the system language;

executing the Active Server Page through the application programming interface to generate the requested Web page; and

providing the requested Web page to a source of the request.

2. (original): A method as recited in claim 1 wherein the user interface template has been compiled into a byte code format and the Active Server Page contains the byte codes.

3. (original): A method as recited in claim 1 wherein the user interface template contains HTML code.

4. (original): A method as recited in claim 1 wherein the user interface template contains logic related to displaying information.

5. (original): A method as recited in claim 1 wherein the Active Server Page includes a plurality of compiled user interface templates.

6. (original): One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 1.

7. (currently amended): A method comprising:
identifying a plurality of user interface templates created using Active Server Page Languages that are each associated with a Web-based application;
compiling, for each of the Active Server Page Languages supported, each of the plurality of user interface templates into a single file containing a plurality of byte codes, wherein the byte codes are capable of being executed by an execution engine that implements an Internet service application programming interface (ISAPI) of the Web-based application that is separate from an application programming interface used to create the Web-based application; and
executing the plurality of byte codes when the Web-based application is executed.

8. (original): A method as recited in claim 7 wherein the plurality of byte codes include callback codes that call into the Web-based application code.

9. (original): A method as recited in claim 7 wherein the plurality of byte codes are executed by an execution engine in a Web server.

10. (original): A method as recited in claim 7 wherein the plurality of byte codes are contained in an Active Server Page.

11. (original): A method as recited in claim 7 wherein the byte codes include logic related to displaying information.

12. (original): One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 7.

13. (currently amended): A method comprising:
creating a plurality of user interface templates associated with a Web-based application, wherein the plurality of user interface templates are each created using an Active Server Page Language and the Web-based application uses an Internet service application programming interface (ISAPI) to implement business logic separately separate from the plurality of user interface templates;
compiling the plurality of user interface templates into a plurality of byte codes prior to execution; and

storing the plurality of byte codes associated with the plurality of user interface templates in a single file, that supports each Active Server Page Language associated with the plurality of user interface templates, wherein the byte codes are capable of being executed by an execution engine in a Web server, the execution engine comprises run time code of the ISAPI that executes the single file derived from the plurality of user interface templates created using Active Server Page Language to generate Web pages using the execution engine being configured to use a system language of the ISAPI that is separate from an application programming interface used for the Web-based application.

14. (original): A method as recited in claim 13 further comprising executing the plurality of byte codes when the Web-based application is executed.

15. (original): A method as recited in claim 13 wherein the plurality of byte codes include callback codes that call into the Web-based application code.

16. (original): A method as recited in claim 13 further comprising executing a portion of the plurality of byte codes when the Web-based application is executed.

17. (original): One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 13.

18. (currently amended): An apparatus comprising:
- a processor and one or more computer-readable memories containing a computer program that is executable by the processor to form:
 - an interface to receive requests for Web pages and to send responses to the received requests; and
 - an execution engine coupled to the interface, wherein the execution engine is configured:
 - to identify an Active Server Page associated with a request for a Web page, wherein the Active Server Page includes a plurality of user interface templates, each of the user interface templates being created using an Active Server Page Language that is separate from an application programming interface used for the Active Server Page; and
 - to identify user interface template information contained in the Active Server Page, wherein the execution engine is further configured to execute the Active Server Page through run time code of ~~an~~ the application programming interface to generate the requested Web page in a system language selected from among Active Server Page Languages supported by the plurality of user interface templates to provide the requested Web page to a source of the request.

19. (previously presented): An apparatus as recited in claim 18 wherein the Active Server Page contains a plurality of byte codes associated with the plurality of user interface templates.

20. (original): An apparatus as recited in claim 19 wherein the execution engine executes the byte codes associated with the request.

21. (currently amended): An apparatus comprising:

means for identifying a plurality of user interface templates, each user interface template being created using an Active Server Page Language that individually implements an Internet Service Application Programming Interface for the associated Active Server Page Language and is associated with a Web-based application that uses a system language that is different from a particular said Internet Service Application Programming Interface for a particular said user interface template;

means for compiling each of the plurality of user interface templates into a single file, that supports multiple Active Server Page Languages that are each associated with a user interface template from the plurality of user interface templates, containing a plurality of byte codes, wherein the plurality of byte codes are capable of being executed by an execution engine that implements an application programming interface of the system language to generate Web pages in the system language from the user interface templates created using one of the Active Server Page Languages; and

means for executing at least a portion of the plurality of byte codes when the Web-based application is executed.

22. (original): An apparatus as recited in claim 21 wherein the byte codes are contained in an Active Server Page.

23. (original): An apparatus as recited in claim 21 wherein the byte codes include logic related to displaying information.

24. (previously presented): One or more computer-readable storage media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:

create a plurality of user interface templates associated with a Web-based application, wherein the plurality of user interface templates are each created using an Active Server Page Language and the Web-based application uses an application programming interface in a system language to implement business logic separately from the plurality of user interface templates;

compile the plurality of user interface templates into a plurality of byte codes of the system language; and

store the plurality of byte codes in a single file, that supports multiple Active Server Page Languages that are associated with the plurality of user interface templates, wherein the byte codes are capable of being executed by a Web server that implements the application programming interface of the system language to generate a Web page.

25. (previously presented): One or more computer-readable storage media as recited in claim 24 wherein the one or more processors further execute at least a portion of the byte codes when the Web-based application is executed.

26. (previously presented): One or more computer-readable storage media as recited in claim 24 wherein the plurality of byte codes include at least one callback code that calls into the Web-based application code.